Rough feature selection for intelligent classifiers

Shen, Qiang

Published in:
Transactions on Rough Sets VII

Publication date:
2007

Citation for published version (APA):
Shen, Q. (2007). Rough feature selection for intelligent classifiers. In J. F. Peters , A. Skowron, V. W. Marek, E. Orowska, R. Sowiski, & W. Ziarko (Eds.), *Transactions on Rough Sets VII: Commemorating the Life and Work of Zdzisaw Pawlak, Part I* (Vol. 4400, pp. 244-255). (Lecture Notes in Computer Science; Vol. 4400). Springer Nature.

Rough Feature Selection for Intelligent Classifiers

Qiang Shen

Department of Computer Science
The University of Wales
Aberystwyth SY23 3DB, UK
qqs@aber.ac.uk

Abstract. The last two decades have seen many powerful classification systems being built for large-scale real-world applications. However, for all their accuracy, one of the persistent obstacles facing these systems is that of data dimensionality. To enable such systems to be effective, a redundancy-removing step is usually required to pre-process the given data. Rough set theory offers a useful, and formal, methodology that can be employed to reduce the dimensionality of datasets. It helps select the most information rich features in a dataset, without transforming the data, all the while attempting to minimise information loss during the selection process. Based on this observation, this paper discusses an approach for semantics-preserving dimensionality reduction, or feature selection, that simplifies domains to aid in developing fuzzy or neural classifiers. Computationally, the approach is highly efficient, relying on simple set operations only. The success of this work is illustrated by applying it to addressing two real-world problems: industrial plant monitoring and medical image analysis.

1 Introduction

Knowledge-based classification systems have been successful in many application areas. However, complex application problems, such as reliable monitoring and diagnosis of industrial plants and trustworthy analysis and comparison of medical images, have emphasised the issue of large numbers of features present in the problem domain, not all of which will be essential for the task at hand. The applicability of most classification systems is often limited by the curse of dimensionality that imposes a ceiling on the complexity of the application domain. A method to allow generation of intelligent classifiers for such application domains is clearly desirable.

Dimensionality reduction is also required to improve the runtime performance of a classifier. For example, in industrial plant monitoring, by requiring less observations per variable, the dimensionality reduced system becomes more compact and its response time decreases. The cost of obtaining data drops accordingly, as fewer connections to instrumentation need be maintained. In the meantime, the overall robustness of the system can increase, since, with fewer instruments, the chances of instrumentation malfunctions leading to spurious readings may be reduced dramatically.

Inspired by such observations, numerous different dimensionality reduction methodologies have been proposed in the literature. Unfortunately, many of them remove redundancy by irretrievably destroying the original meaning of the data given for learning. This significantly reduces, if not completely loses, the potential expressive power

of the classification systems for computing with clear semantics. This, in turn, leads to a lack of trust in such systems, while such trust is usually critical for the systems to be taken up by end users.

The work on rough set theory [7] offers an alternative, and formal, methodology (amongst many other possible applications, e.g. [6, 8]) that can be employed to reduce the dimensionality of datasets, as a preprocessing step to assist the development of any type of classifiers via learning from data. It helps select the most information rich features in a dataset, without transforming the data, all the while attempting to minimise information loss during the selection process [14]. Computationally, the approach is highly efficient, relying on simple set operations, which makes it suitable as a preprocessor for techniques that are much more complex. Unlike statistical correlation-reducing approaches [1], it requires no human input or intervention and retains the semantics of the original data.

Combined with an intelligent classification system built by, say, a fuzzy system or a neural network, the feature selection approach based on rough set theory can not only retain the descriptive power of the overall classifier, but also allow simplified system structure. This helps enhance the interoperability and understandability of the resultant systems and their reasoning. Drawing on the initial results previously presented in [12–14], this paper demonstrates the applicability of this approach in supporting transparent fuzzy or neural classifiers, with respect to two distinct application domains.

The remainder of this paper is structured as follows. The rough set-assisted feature selection mechanism is summarised in section 2 for self-containedness. This is followed by an illustration of the two example applications, demonstrating how different classification tasks can benefit from rough set-assisted semantics-preserving dimensionality reduction. The paper is concluded in section 5, with interesting further work pointed.

2 Feature Selection

This section shows the basic ideas of rough sets [7] that are relevant to the present work and describes an efficient computational algorithm, named Rough Set Attribute Reduction (RSAR), for feature selection.

2.1 Rough Sets

A rough set is an approximation of a vague concept by a pair of precise concepts, called lower and upper approximations. The lower approximation is a description of the domain objects which are known with absolute certainty to belong to the subset of interest, whereas the upper approximation is a description of the objects which possibly belong to the subset.

Rough sets have been employed to remove redundant conditional attributes from discrete-valued datasets, while retaining their information content. Central to this work is the concept of indiscernibility. Without losing generality, let $I = (U, A)$ be an information system, where U is a non-empty set of finite objects (the universe of discourse), and A is a non-empty finite set of variables such that $a : U \rightarrow V_a \; \forall a \in A$, V_a being the value set of variable a. In building a classification system, for example, $A = \{C \cup D\}$

where C is the set of input features and D is the set of class indices. Here, a class index $d \in D$ is itself a variable $d : U \rightarrow \{0, 1\}$ such that for $a \in U, d(a) = 1$ if a has class d and $d(a) = 0$ otherwise.

With any $P \subseteq A$ there is an associated equivalence relation $IND(P)$:

$$IND(P) = \{(x, y) \in U \times U \mid \forall\, a \in P,\ a(x) = a(y)\} \tag{1}$$

Note that this corresponds to the equivalence relation for which two objects are equivalent if and only if they have vectors of attribute values for the attributes in P. The partition of U, determined by ***IND(P)*** is denoted U/P, which is simply the set of equivalence classes generated by $IND(P)$.

If $(x, y) \in IND(P)$, then x and y are indiscernible by features in P. The equivalence classes of the P-indiscernibility relation are denoted $[x]_P$. Let $X \subseteq U$, the P-*lower* and P-*upper* approximations of a classical crisp set are respectively defined as:

$$\underline{P}X = \{x \mid [x]_P \subseteq X\} \tag{2}$$

$$\overline{P}X = \{x \mid [x]_P \cap X \neq \emptyset\} \tag{3}$$

Let *P* and *Q* be subsets of A, then the important concept of *positive region* is defined as:

$$POS_P(Q) = \bigcup_{X \in U/Q} \underline{P}X \tag{4}$$

For tasks like classification with feature patterns, the positive region contains all objects of U that can be classified into classes of U/Q using the knowledge conveyed by the features of *P*.

2.2 Feature Dependency and Significance

The important issue here is to discover dependencies of object classes upon given features. Intuitively, a set of classes *Q* depends totally on a set of features *P*, denoted *P* $\Rightarrow$ *Q*, if all class indices from *Q* are uniquely determined by values of features from *P*. Dependency can be measured in the following way [14]:

For *P,Q* $\subseteq$ *A*, *Q* depends on *P* in a degree *k* ($0 \leq k \leq 1$), denoted $P \Rightarrow_k Q$, if

$$k = \gamma_P(Q) = \frac{|POS_P(Q)|}{|U|} \tag{5}$$

where $|S|$ stands for the cardinality of set S.
If *k* = 1, *Q* depends totally on *P*; if $0 < k < 1$, *Q* depends partially (in a degree *k*) on *P*; and if *k* = 0, *Q* does not depend on *P*.

By calculating the change in dependency when a feature is removed from the set of considered possible features, an estimate of the significance of that feature can be obtained. The higher the change in dependency, the more significant the feature is. If the significance is 0, then the feature is dispensable. More formally, given *P,Q* and a feature *x* $\in$ *P*, the significance of feature x upon *Q* is defined by

$$\sigma_P(Q, x) = \gamma_P(Q) - \gamma_{P-\{x\}}(Q) \tag{6}$$

2.3 Feature Selection Algorithm

The selection of features is achieved by reducing the dimensionality of a given feature set, without destroying the meaning conveyed by the individual features selected. This is, in turn, achieved by comparing equivalence relations generated by sets of features with regard to the underlying object classes, in the context of classification.

Features are removed so that the reduced set will provide the same quality of classification as the original. For easy reference, the concept of *retainer* is introduced as a subset *R* of the initial feature set C such that $\gamma_R(D) = \gamma_C(D)$. A minimal retainer is termed a *reduct* in the literature [9]. That is, a further removal of any feature from a reduct will make it violate the constraint $\gamma_R(D) = \gamma_C(D)$.

Thus, a given dataset may have many feature retainers, and the collection of all retainers is denoted by

$$R = \{X \mid X \subseteq C, \gamma_X(D) = \gamma_C(D)\} \tag{7}$$

The intersection of all the sets in *R* is called the *core*, the elements of which are those features that cannot be eliminated without introducing more contradictions to the representation of the dataset. Clearly, for feature selection, an attempt is to be made to locate a minimal retainer, or a single reduct, $R_{min} \subseteq R$:

$$R_{min} = \{X \mid X \in R,\ \forall Y \in R, |X| \leq |Y|\} \tag{8}$$

A basic way of achieving this is to calculate the dependencies of all possible subsets of C. Any subset X with $\gamma_X(D) = 1$ is a retainer; the smallest subset with this property is a reduct. However, for large datasets with a large feature set this method is impractical and an alternative strategy is required.

1. $R \leftarrow \{\}$
2. do
3. $\quad T \leftarrow R$
4. $\quad \forall x \in (C - R)$
5. $\qquad$ if $\gamma_{R \cup \{x\}}(D) > \gamma_T(D)$
6. $\qquad\quad T \leftarrow R \cup \{x\}$
7. $\quad R \leftarrow T$
8. until $\gamma_R(D) = \gamma_C(D)$
9. return R

Fig. 1. The RSAR feature selection algorithm.

The RSAR feature selection algorithm given in Figure 1 attempts to calculate a reduct without exhaustively generating all possible subsets. It starts off with an empty set and adds in turn, one at a time, those features that result in the greatest increase in $\gamma_P(Q)$, until the maximum possible value of $\gamma_P(Q)$, usually 1, results for the given dataset. Note that this method does not always generate a *minimal* retainer (or reduct),

as $\gamma_P(Q)$ is not a perfect heuristic. However, it does result in a close-to-minimal retainer, which is still useful in greatly reducing feature set dimensionality. It is also worth noting that one way to guarantee the generation of a reduct is to apply RSAR in conjunction with a selection strategy that works in reverse order (i.e., starting with a full set of features and then deleting one at a time). Nevertheless, such an approach has a significant practical limit when the original feature set is of a high dimensionality.

RSAR works in a greedy manner, not compromising with a set of features that contains a large part of the information of the initial set. It attempts to reduce the feature set without loss of information significant to solving the problem at hand. The way it works is clearly dependent upon features being represented in nominal values. However, this does not necessarily give rise to problems in the use of the overall classification system which includes such a feature selection preprocessor. This is because the real feature values are only required to be temporarily discretised for feature selection itself. The classifier will use the original real-valued features directly. In this regard, it is independent of the classification methods adopted. When used in conjunction with an explicit descriptive classifier, the resulting system will be defined in terms of only the significant features of the data, retaining the desirable transparency. The training process is accelerated, while the runtime operation of the system is sped up since fewer attributes are required.

3 Application I: Industrial Plant Monitoring

This application concerns the task of monitoring a water treatment plant [14]. To illustrate the generality of the presented approach and its, involving the use of a fuzzy system based classifier. This domain was chosen because of its realism. A large plant is likely to involve a number of similar features, not all of which will be essential in determining the operational status. Interrelations between features are unavoidable as the plant is a single system with interconnections, leading to a fair degree of redundancy.

3.1 Problem Case

The Water Treatment dataset comprises a set of historical data obtained over a period of 521 days, with one series of measurements per day. Thirty eight different feature values are measured per day, with one set of such measurements forming one datum. All measurements are real-valued. The goal is to implement a fuzzy classification system that, given this dataset of past measurements and without the benefit of an expert in the field at hand, will classify the plant's status and produce human comprehensible explanations of the monitoring results.

The thirty eight features account for the following five aspects of the water treatment plant's operation (see Figure 2 for an illustration of this): input to plant; input to primary settler; input to secondary settler; output from plant; and overall plant performance. The operational state of the plant is represented by a boolean categorisation representing the detection of a fault. The point is to draw the operator's attention to an impending fault.

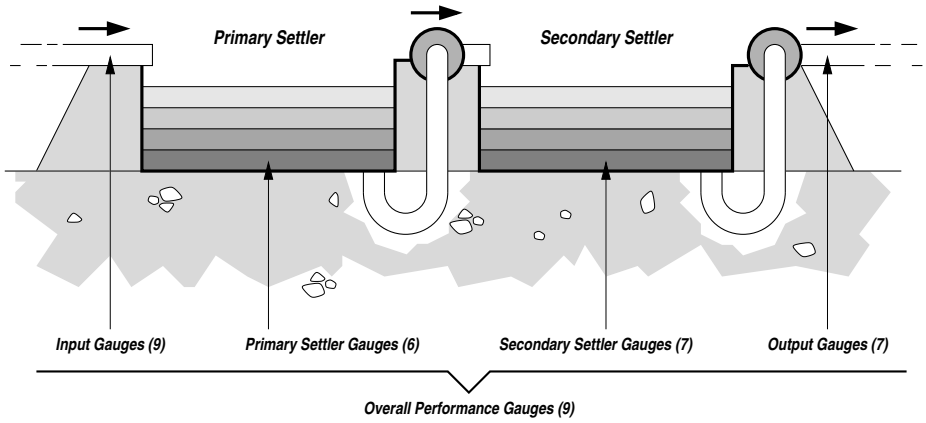


Fig. 2. Schematic diagram of the water treatment plant, indicating the number of measurements sampled at various points.

3.2 Fuzzy Classifier

In this experimental study, to obtain a system that will entail classification of the plant's operating status, the fuzzy induction algorithm first reported in [3] is used. This is adopted simply due to the availability of its software implementation; any other fuzzy rule induction method may be utilised as an alternative for classifier building. The resulting classification system is represented in a set of fuzzy production rules. For the sake of completeness, an outline of the induction algorithm employed is given below.

The algorithm generates a hyperplane of candidate fuzzy rules by fuzzifying the entire training dataset using all permutations of the input features. Thus, a system with M inputs, each of which has a domain fuzzified by f_j fuzzy sets ($1 \leq j \leq M$), the hyperplane is fuzzified into $\prod_{j=1}^{M} f_j$ M-dimensional clusters, each representing one vector of rule preconditions. Each cluster $\underline{p} = \langle D^1, D^2, \ldots, D^M \rangle$ may lead to a fuzzy rule, provided that the given dataset supports it.

To obtain a measure of what classification applies to a cluster, fuzzy min-max composition is used. The input feature pattern of each example object is fuzzified according to the fuzzy sets $\{\mu_{D^1}, \mu_{D^2}, \ldots, \mu_{D^M}\}$ that make up cluster $\underline{p}$. For each object $\underline{x} = \langle x_1, x_2, \ldots, x_M \rangle$, the following t-norm of it, with respect to cluster $\underline{p}$ and classification c, is calculated:

$$T_c^{\underline{p}} \underline{x} = \min \left(\mu_{D^1}(x_1), \mu_{D^2}(x_2), \ldots, \mu_{D^M}(x_M) \right) \tag{9}$$

Furthermore, the maximum of all t-norms with respect to $\underline{p}$ and c is then calculated and this is dubbed an s-norm:

$$S_c^{\underline{p}} = \max \left\{ T_c^{\underline{p}} \underline{x} \mid \underline{x} \in C_c \right\} \tag{10}$$

where C_c is the set of all examples that can be classified as c. This is iterated over all possible classifications to provide a full indication of how well each cluster applies to each classification.

A cluster generates at most one classification rule. The rule's preconditions are the cluster's M co-ordinate fuzzy sets connected conjunctively. The conclusion is the classification attached to the cluster. Since there may be s-norms for more than one classification, it is necessary to decide on one classification for each of the clusters. Such contradictions are resolved by using the *uncertainty margin*, ε ($0 \leq \varepsilon < 1$). An s-norm assigns its classification on its cluster if and only if it is greater by at least ε than all other s-norms for that cluster. If this is not the case, the cluster is considered undecidable and no rule is generated. The uncertainty margin introduces a trade-off in the rule generation process between the size and the accuracy of the resulting classification. In general, the higher ε is, the less rules are generated, but classification error may increase. A fuller treatment of this algorithm in use for descriptive learning can be found in [3].

3.3 Results

Running the RSAR algorithm on the Water Treatment dataset provided a significant reduction, with merely two features selected from the total of 38. Testing on previously unseen data resulted in a classification accuracy of 97.1%, using the fuzzy model generated by the above-mentioned rule induction method.

A comparison against a widely recognised benchmark method should help in establishing the success of the system. C4.5 [10] is a widely accepted and powerful algorithm that provides a good benchmark [5] for learning from data. The decision trees it generates allow for rapid and efficient interpretation. Yet, C4.5's decision tree for the present problem involves a total of three attributes from the dataset, as opposed to two chosen by the RSAR algorithm. In terms of classification performance, C4.5 obtains a compatible accuracy of around 96.8%.

Note that training a fuzzy system on all 38 features would be computationally prohibitive with the adopted learning algorithm. As stated previously, the benefits do not limit themselves to the learning phase; they extend to the runtime use of the learned classifier. By reducing the dimensionality of the data, the dimensionality of the ruleset is also reduced. This results in fewer measured features, which is very important for dynamic systems where observables are often restricted. This in turn leads to fewer connections to instrumentation and faster system responses in emergencies. Both of which are important to the problem domain.

The most important benefit of using RSAR is, however, derived from its conjunctive use with the linguistically expressive fuzzy system. With the learned rules, it can provide explanations of its reasoning to the operator. This leads to increased trust in the system, as its alarms can be understood meaningfully. A classification system consisting of rules involving 38 features, even though they are all directly measurable and hence individually interpretable, is very difficult to understand, whilst one involving only two features is very easy to interpret.

4 Application II: Medical Image Analysis

Comparing normal and abnormal blood vessel structures, via the analysis of cell images, plays an important role in pathology and medicine [12]. This forms the focus of

this application, analysing medical images by the use of a neural network based image classifier that is supported by RSAR.

4.1 Problem Case

Central to this analysis is the capture of the underlying features of the cell images. Many feature extraction methods are available to yield various kinds of characteristic descriptions of a given image. However, little knowledge is available as to what features may be most helpful to provide the discrimination power between normal and abnormal cells and between their types, while it is computationally impractical to generate many features and then to perform classification based on these features for rapid diagnosis. Generating a good number of features and selecting from them the most informative ones off-line, and then using those selected on-line is the usual way to avoid this difficulty. Importantly, the features produced ought to have an embedded meaning and such meaning should not be altered during the selection process. Therefore, this problem presents a challenging case to test the potential of RSAR.

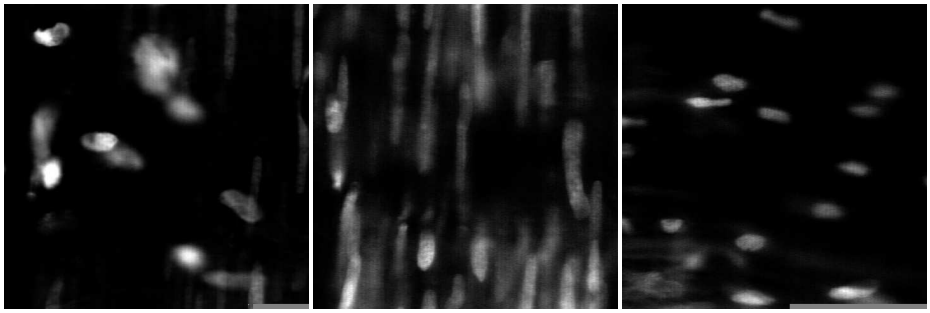
The samples of subcutaneous blood vessels used in this work were taken from patients suffering critical limb ischaemia immediately after leg amputation. The level of amputation was always selected to be in a non-ischaemic area. The vessel segments obtained from this area represented internal proximal (normal) arteries, whilst the distal portion of the limb represented ischaemic (abnormal) ones. Images were collected using an inverted microscope, producing an image database of 318 cell images, each sized 512×512 pixels with grey levels ranging from 0 to 255. Examples of the three types of cell image taken from non-ischaemic, and those from ischaemic, resistance arteries are shown in Figure 3. Note that many of these images seem rather similar to the eye. It is therefore a difficult task for visual inspection and classification.

4.2 Neural Network Classifier

In this work, each image classifier is implemented using a traditional multi-layer feed-forward artificial neural network (MFNN). To capture and represent many possible and essential characteristics of a given image, fractal models [4] are used. Note that, although these particular techniques are herein adopted to perform their respective task, the work described does not rely on them, but is generally applicable when other classification and feature extraction methods are employed.

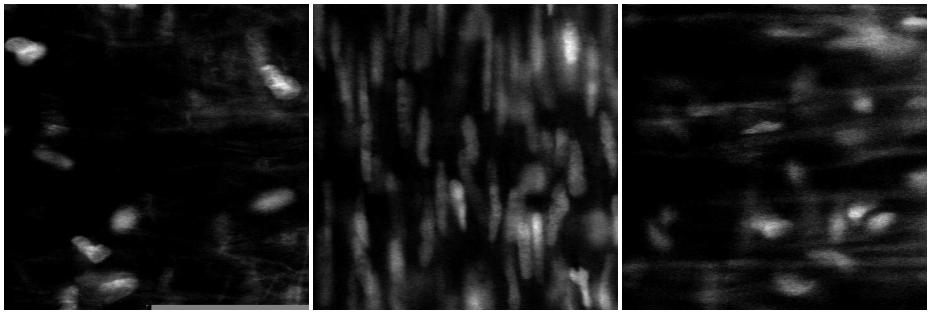
An MFNN-based classifier accomplishes classification by mapping input feature patterns onto their underlying image classes. The design of each MFNN classifier used for the present work is specified as follows. The number of nodes in its input layer is set to that of the dimensionality of the given feature set (before or after feature reduction), and the number of nodes within its output layer is set to the number of underlying classes of interest. The internal structure of the network is designed to be flexible and may contain one or two hidden layers.

The training of the classifier is essential to its runtime performance, and is here carried out using the back-propagation algorithm [11]. For this, feature patterns that represent different images, coupled with their respective underlying image class indices, are selected as the training data, with the input features being normalised into the range



(1) adventitial (2) SMC (3) endothelial

(a) from proximal, non-ischaemic blood vessels



(1) adventitial (2) SMC (3) endothelial

(b) from distal, ischaemic blood vessels

Fig. 3. Section cell images, where the first, second and third columns respectively show adventitial, smooth muscle and endothelial cells in proximal non-ischaemic and distal ischaemic subcutaneous blood vessels, taken from a human lower limb.

of 0 to 1. Here, each feature pattern consists of 9 fractal features (including 5 isotropic fractals measured on the top five finest resolutions and 4 directional fractals [12]) and the mean and standard deviation (STD), with their reference numbers listed in Table 1. Note that when applying the trained classifier, only those features selected during the learning phase are required to be extracted and that no discretisation is needed but real-valued features are directly fed to the classifier.

Feature No.	Feature Meaning	Feature No.	Feature Meaning
1	0° direction	7	3rd finest resolution
2	45° direction	8	4th finest resolution
3	90° direction	9	5th finest resolution
4	135° direction	10	Mean
5	Finest resolution	11	STD
6	2nd finest resolution		

Table 1. Features and their reference number.

4.3 Results

Eighty-five images selected from the image database are used for training and the remaining 233 images are employed for testing. For simplicity, only MFNNs with one hidden layer are considered.

Table 2 lists the results of using RSAR and the original full set of features. The error rate of using the five selected features is lower than that of using the full feature set. This improvement of performance is obtained by a structurally much simpler network of 10 hidden nodes, as opposed to the classifier that requires 24 hidden nodes to achieve the optimal learning. This is indicative of the power of RSAR in helping reduce not only redundant feature measures but also the noise associated with such measurement. Also, the classifier using those five RSAR-selected features considerably outperforms those using five randomly selected features, with the average error of the latter reaching 19.1%.

Method	Dimensionality	Features	Structure	Error
Rough	5	1,4,9,10,11	$5\times10 + 10\times6$	7.55%
Original	11	1,2,3,4,5,6,7,8,9,10,11	$11\times24 + 24\times6$	9.44%

Table 2. Results of using rough-selected and the original full set of features.

Again, a comparison against a widely recognised benchmark method should help reflect the success of the system. For this, the results of rough feature selection are systematically compared to those obtained via the use of Principal Component Analysis (PCA) [1], as summarised in Table 3. Note that PCA is perhaps the most adopted dimensionality reduction technique. Although efficient, it irreversibly destroys the underlying semantics of the feature set. Therefore, in this table, for the results of using PCA, feature number $i, i \in \{1, 2, ..., 11\}$, stands for the ith principal component, i.e. the transformed feature that is corresponding to the ith largest variance.

The advantages of using RSAR are clear. Of the same dimensionality (i.e., 5), the classifier using the features selected by the rough set approach has a substantially higher classification accuracy, and this is achieved via a considerably simpler neural network.

Method	Dimensionality	Features	Structure	Error
Rough	**5**	**1,4,9,10,11**	$\mathbf{5{\times}10 + 10{\times}6}$	**7.7%**
PCA	1	1	$1{\times}12 + 12{\times}6$	57.1%
	2	1,2	$2{\times}12 + 12{\times}6$	32.2%
	3	1,2,3	$3{\times}12 + 12{\times}6$	31.3%
	4	1,2,3,4	$4{\times}24 + 24{\times}6$	28.8%
	5	**1,2,3,4,5**	$\mathbf{5{\times}20 + 20{\times}6}$	**18.9%**
	6	1,2,3,4,5,6	$6{\times}18 + 18{\times}6$	15.4%
	7	1,2,3,4,5,6,7	$7{\times}24 + 24{\times}6$	11.6%
	8	1,2,3,4,5,6,7,8	$8{\times}24 + 24{\times}6$	13.7%
	9	1,2,3,4,5,6,7,8,9	$9{\times}12 + 12{\times}6$	9.9%
	10	1,2,3,4,5,6,7,8,9,10	$10{\times}20 + 20{\times}6$	7.3%
	11	1,2,3,4,5,6,7,8,9,10,11	$11{\times}8 + 8{\times}6$	7.3%

Table 3. Results of using rough and PCA-selected features.

When increasing the dimensionality of principal features, the error rate generally gets reduced, but the classifier generally underperforms until almost the full set of principal features is used. The overall structural complexity of all these classifiers are more complex than that of the classifier using the five RSAR-selected features. In addition, the use of those classifiers that use PCA-selected features would require many more feature measurements to achieve comparable classification results.

5 Conclusion

It is well-known that the applicability of most intelligent classification approaches is limited by the curse of dimensionality, which imposes a ceiling on the complexity of the application domain. This paper has demonstrated an effective approach to semantics-preserving dimensionality reduction by exploiting the basic ideas of rough set theory. Such a feature selection tool makes learned classifiers much more transparent and comprehensible to humans, who have inherent trouble understanding high-dimensionality domains, in addition to being able to lessen the obstacles of the dimensionality ceiling.

In summary, Rough Set Attribute Reduction (RSAR) selects the most information rich attributes in a dataset, without transforming the data, all the while attempting to minimise information loss as regards the classification task at hand. When employed by an intelligent classification system (be it a fuzzy system or neural network), by simplifying the problem domain, RSAR helps enhance the transparency and maintain the accuracy of the classifier. With relatively simple system structures, the examination of the quality of the results inferred by the use of such classifiers is made easy. This has been demonstrated in applications to two rather different problem domains, with very promising results.

Although RSAR has been used as a dataset pre-processor with much success, it is reliant upon a crisp dataset. Important information (for choosing the optimal features) may be lost as a result of required boolean discretisation of the underlying numerical

features. Further advances have recently been made in proposing a feature selection technique that employs a hybrid variant of rough sets, the fuzzy-rough sets [2], to avoid this information loss [15]. Whilst this is out of the scope of this paper, it is interesting to point out that initial experimental results, of applying this improved version to the problem of industrial plant monitoring, have shown that fuzzy-rough feature selection is more powerful than many conventional approaches, including entropy-based, PCA-based and random-based methods.

Acknowledgments

The author is very grateful to many of his colleagues, especially to Alexios Chouchoulas, Richard Jensen and Changjing Shang, for their contribution in the work, whilst taking full responsibility for the views expressed in this paper.

References

1. Devijver, P. and Kittler, J. Pattern Recognition: a Statistical Approach. Prentice Hall, 1982.
2. Dudois, D. and Prade, H. Putting rough sets and fuzzy sets together. In: R. Slowinski (Ed.), Intelligent Decision Support. Kluwer Academic Publishing, pages 203–232, 1992.
3. Lozowski, A. Cholewo, T. and Zurada, J. Crisp rule extraction from perceptron network classifiers. In *Proceedings of International Conference on Neural Networks*, volume Plenary, Panel and Special Sessions, pages 94–99, 1996.
4. Mandelbrot, B. The Fractal Geometry of Nature. San Francisco: Freeman, 1982.
5. Mitchell, T. *Machine Learning.* McGraw-Hill (1997).
6. Orlowska, E. Incomplete Information: Rough Set Analysis. Springer Verlag, 1997.
7. Pawlak, Z. *Rough Sets: Theoretical Aspects of Reasoning About Data*. Kluwer Academic Publishers, Dordrecht (1991).
8. Peters, J. and Skowron, A. A rough set approach to knowledge discovery. *International Journal of Intelligent Systems* **17** (2002) 109-112.
9. Pawlak, Z. and Skowron, A. Rough set rudiments. *Bulletin of International Rough Set Society* **3(4)** (2000) 43-47.
10. Quinlan, J. R. *C4.5: Programs for Machine Learning*. The Morgan Kaufmann Series in Machine Learning. Morgan Kaufmann Publishers (1993).
11. Rumelhant, D. Hinton, E. and Williams, R. Learning internal representations by error propagating. In: E. Rumelhant and J. McClelland (Eds.), Parallel Distributed Processing. MIT Press, 1986.
12. Shang, C. and Shen, Q. Rough feature selection for neural network based image classification. *International Journal of Image and Graphics* **2** (2002) 541-555.
13. Shen, Q. Semantics-preserving dimensionality reduction in intelligent modelling. In: . Lawry, J. Shanahan and A. Ralescu (Eds.), Modelling with Words. Springer, 2003.
14. Shen, Q. and Chouchoulas, A. A fuzzy-rough approach for generating classification rules. *Pattern Recognition*, **35** (2002) 341–354.
15. Shen, Q. and Jensen, R. Selecting informative features with fuzzy-rough sets and its application for complex systems monitoring. *Pattern Recognition*, **37** (2004) 1351–1363.